

La structure événement : du principal au sous-programme !

Par David Charalampous, UMR 8214 ISMO, bat 520, Orsay
(Ile de France)

Journée AlpesView, Mercredi 08 décembre 2021



Déroulement des évènements :

I. Contexte de Recherche

1. Expérience VULCAIM : mesure de temps de vol
2. Utilisation dynamique de la structure évènement
3. Utilisation de table dynamique

II. Introduction à la structure évènement

1. L'évènement dans l'IHM
2. Abonnement statique
3. Abonnement Dynamique

III. De l'évènement statique à dynamique

1. Marre de réécrire le même code!
2. Comment j'ai utilisé l'abonnement dynamique
3. Du VI au sous vi :

Conclusion

Annexe

1. Les 18 commandements de l'évènement
2. Gestion évènements utilisateurs
3. L'inspecteur d'évènements

I. Contexte de Recherche

1. Exp rience VULCAIM
2. La mesure de temps de vol
3. Utilisation de table dynamique

I. Contexte de Recherche

Expérience VULCAIM au laboratoire ISMO

Laboratoire ISMO, UMR 8214 :

Institut des Sciences Moléculaires d'Orsay

159 personnes

6 équipes de recherches dans les domaines:
nanobio, nanophysique, Interactions
rayonnement matière atomes-surface-
interfaces et molécules

Equipe SYSTEMAE de l'ISMO

SYSTEMes Moléculaires, Astrophysique et Environnement

23 permanents, 14 doctorants/post-doc

4 thèmes d'études sur des processus
fondamentaux en chimie-physique
moléculaire, intra et intermoléculaires,
signatures spectrales.

Outils expérimentaux utilisés par SYSTEMAE :

- Lasers
- Rayonnement Synchrotron (à SOLEIL, Gif sur Yvette)
- Faisceaux d'ions
 - ⇒ Large gamme d'énergies, des rotations moléculaires aux énergies des états électroniques très excités
- Montages expérimentaux
 - ⇒ Spectroscopie
 - ⇒ Dynamique des systèmes étudiés.

Approches théoriques pour la modélisation et compréhension des processus en jeu

- Thermodynamique statistique
- Chimie quantique
- Dynamique vibrationnelle

I. Contexte de Recherche

Expérience VULCAIM

Banc expérimental VULCAIM :

Reconstitution d'une partie de la photochimie VUV (<200 nm) de la matière diluée dans les milieux astrophysiques.

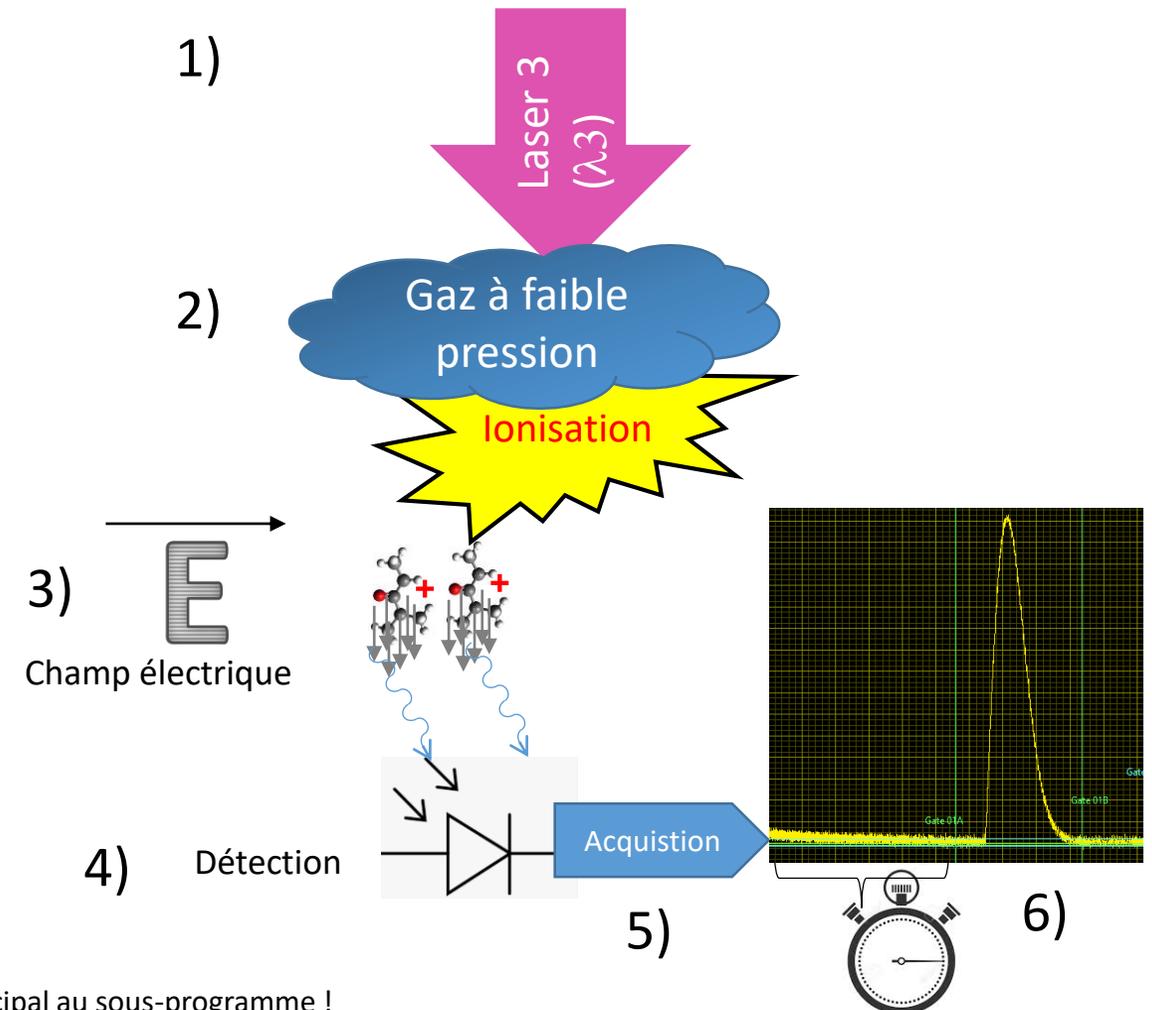
supervisé par Bérenger Gans,
Chargé de recherche



Processus expérimental :

- 1) Combinaison de deux faisceau UV et visible en VUV (λ_3)
- 2) Ionisation du milieu dilué
- 3) Accélération des espèces ionisées
- 4) Détection
- 5) Acquisition
- 6) Mesure de temps de vols
=> spectre de masse

Laser 1 (λ_1 mobile) + Laser 2 (λ_2 fixe)



I. Contexte de Recherche

Mesure du temps de vol, et spectroscopie

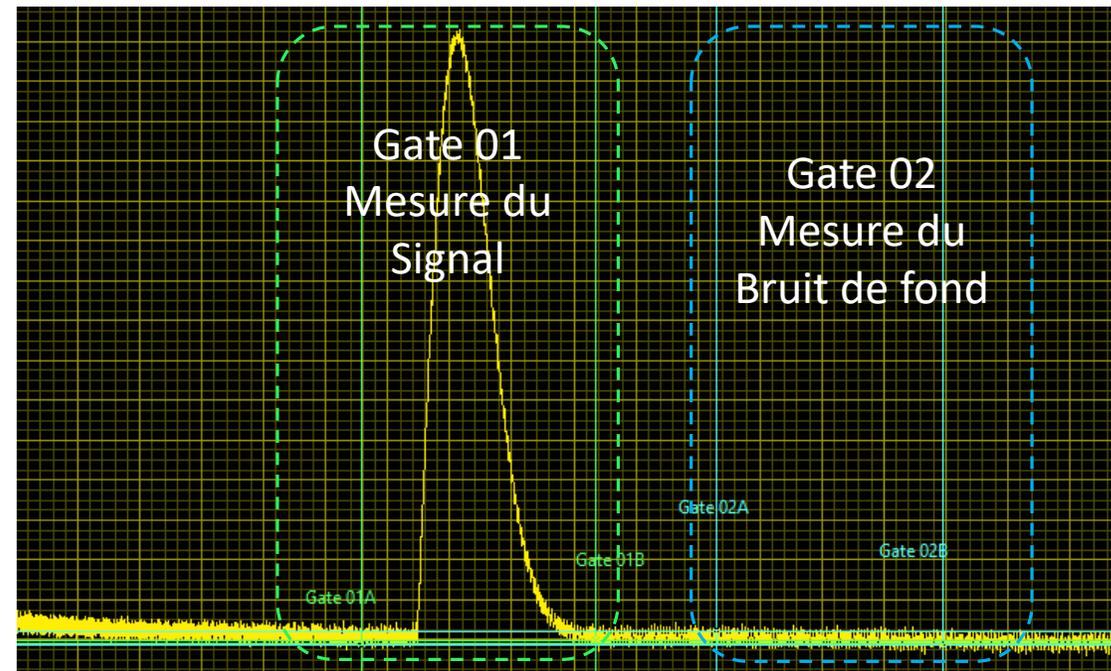
Position temporelle de la gate :

caractéristique du temps de vol
 rapport $m/q \Rightarrow$ poids de l'espèce détectée

Gates :

Pour chaque espèce chimique ionisée il faut une gate de mesure de signal.

Signal intégré puis moyenné par longueur d'onde \Rightarrow construction d'un spectre par espèce.



Nécessité de créer un tableau de gate dynamique :
 ajouts et suppressions aisés, blocage déblocage des curseurs, lier les curseurs couplés en gate, couleurs des gates, nom affichés.....

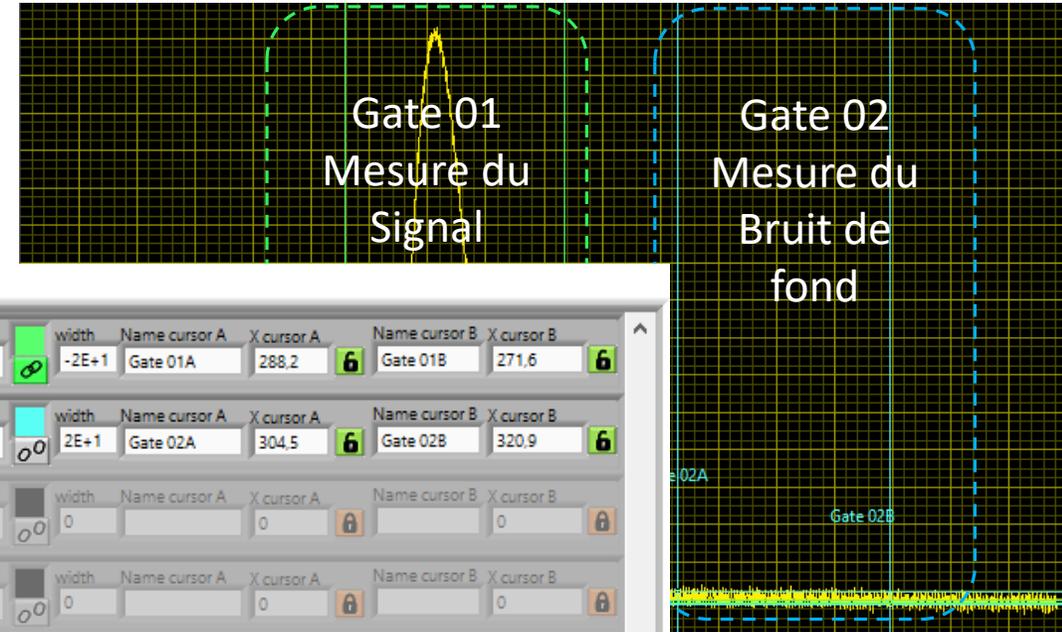
Fonctionnalités uniquement disponible via une structure événement !!!

I. Contexte de Recherche

Tables dynamique

Attributs des Gates :

- couleurs, autorisation de glissement d'un des deux curseurs
- Conservations des attributs des curseurs lors du glissement : tous sauf positions
- Attributs modifiables sur graphe : position étiquette de texte et style du curseur



GatesChannel 1

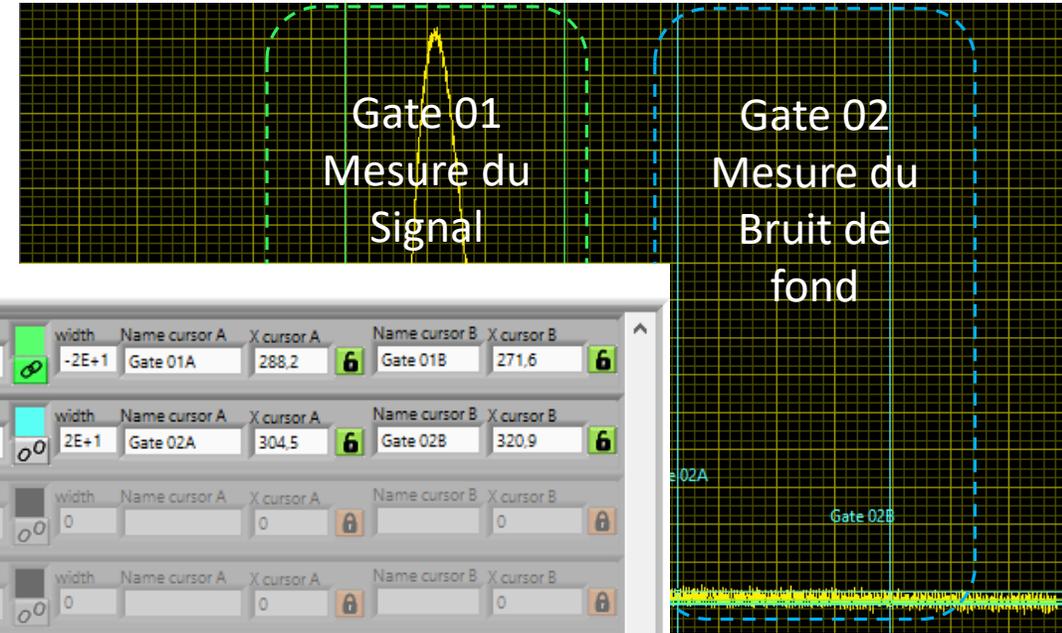
GateName	width	Name cursor A	X cursor A	Name cursor B	X cursor B
- Gate 01	 -2E+1	Gate 01A	288,2	Gate 01B	271,6
- Gate 02	 2E+1	Gate 02A	304,5	Gate 02B	320,9
-	 0		0		0
-	 0		0		0
-	 0		0		0
-	 0		0		0
-	 0		0		0
-	 0		0		0
-	 0		0		0
-	 0		0		0

I. Contexte de Recherche

Tables dynamique

Attributs des Gates :

- couleurs, autorisation de glissement d'un des deux curseurs
- Conservations des attributs des curseurs lors du glissement : tous sauf positions
- Attributs modifiables sur graphe : position étiquette de texte et style du curseur



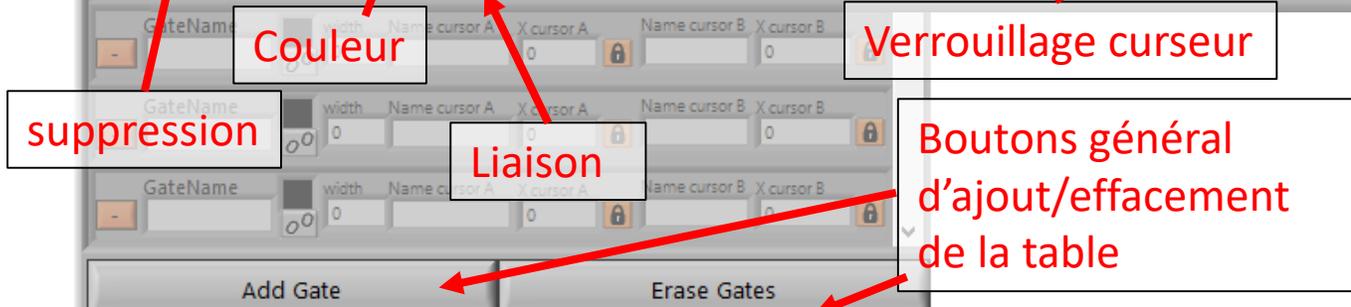
GatesChannel 1

GateName	width	Name cursor A	X cursor A	Name cursor B	X cursor B
Gate 01	-2E+1	Gate 01A	288,2	Gate 01B	271,6
Gate 02	2E+1	Gate 02A	304,5	Gate 02B	320,9
	0		0		0
	0		0		0

- Gate 01: width -2E+1, Gate 01A (288,2), Gate 01B (271,6)
 - Gate 02: width 2E+1, Gate 02A (304,5), Gate 02B (320,9)

Add Gate | Erase Gates

Attributs modifiables par évènement



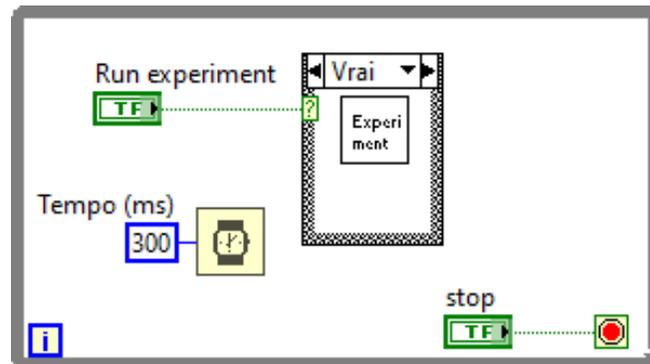
II. Introduction   la structure  v nement

1. L' v nement dans l'IHM
2. Les 18 commandements de l' v nement selon National Instrument
3. Abonnement statique
4. Abonnement Dynamique

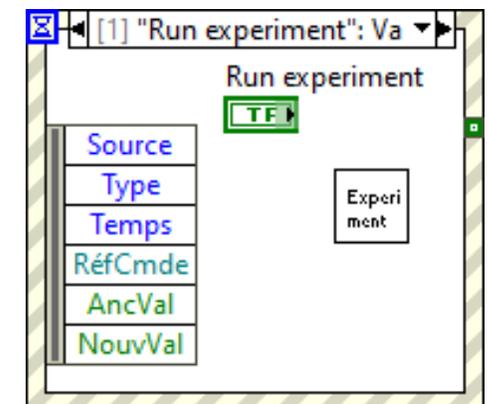
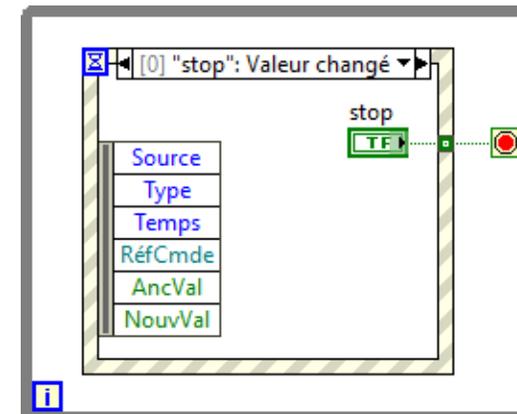
II. Introduction   la structure  v nement

L' v nement dans l'IHM, Interface Humain Machine

- Sans Structure  v nement : examen r gulier des variables ou « polling »
 - ⇒ Temporisation :
 - Tempo faible => consommation de ressource  lev e
 - Tempo  lev e => risque de d synchronisation de t ches et/ou temps d'attente plus important



- Structure  v nement :
 - ⇒ Agit comme une interruption, l'action se produit au moment d'une action utilisateur : clic souris, changement de valeur, curseur d plac , glissement de souris sur objet, ...
 - ⇒ Pas de temps mort
 - ⇒ Fonctionnalit s de l'interface accrue

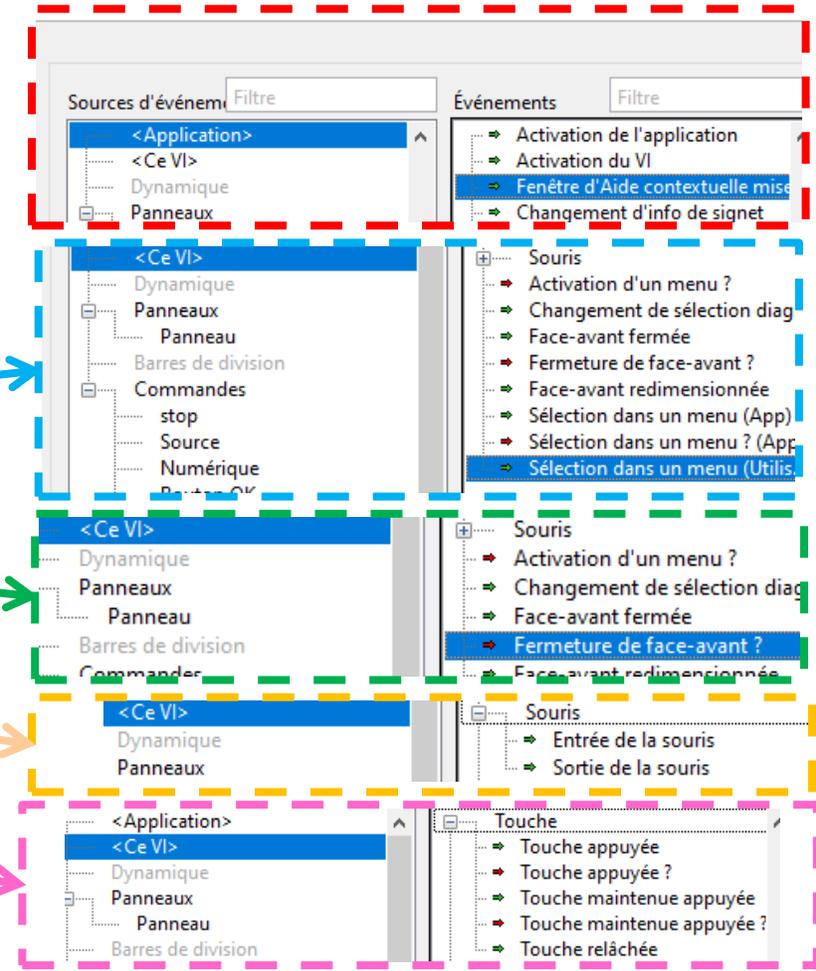


1. Introduction   la structure  v nement

Abonnement statique   des  v nements

- Exemples de fonctionnalit s apport es par la structure  v nements :

- **Acc s   l'aide contextuelle**
- **Acc s au menus**
- **Fermeture Face avant**
- **Entr e de souris**
- **Touche appuy e**
- Etc....



I. Introduction   la structure  v nement

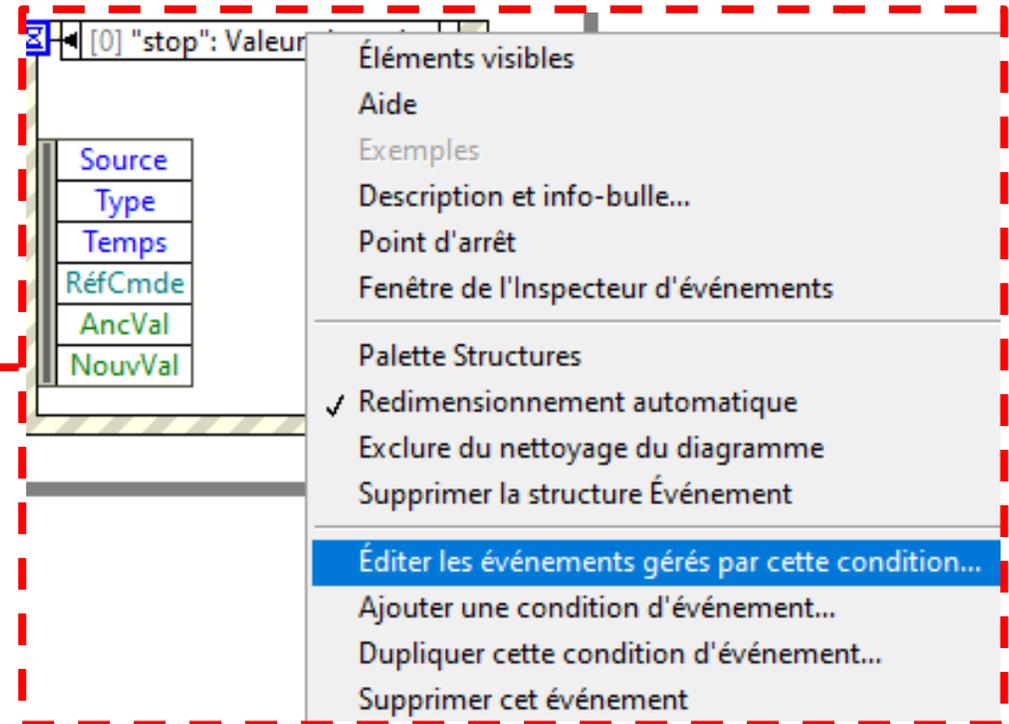
Abonnement statique   des  v nements

Extrait de l'aide de Labview 2018 sur les **abonnement statique** :

« Vous pouvez configurer une structure  v nement afin de sp cifier quels  v nements d'interface utilisateur sur la face-avant d'un VI vous souhaitez g rer dans chaque condition de la structure  v nement.

Utilisez la bo te de dialogue  diter les  v nements pour configurer une structure  v nement afin qu'elle g re un  v nement auquel vous vous  tes abonn  de man re statique.

Remarque Vous ne pouvez pas configurer de man re statique une structure  v nement pour qu'elle traite les  v nements de la face-avant d'un autre VI. »



I. Introduction à la structure événement

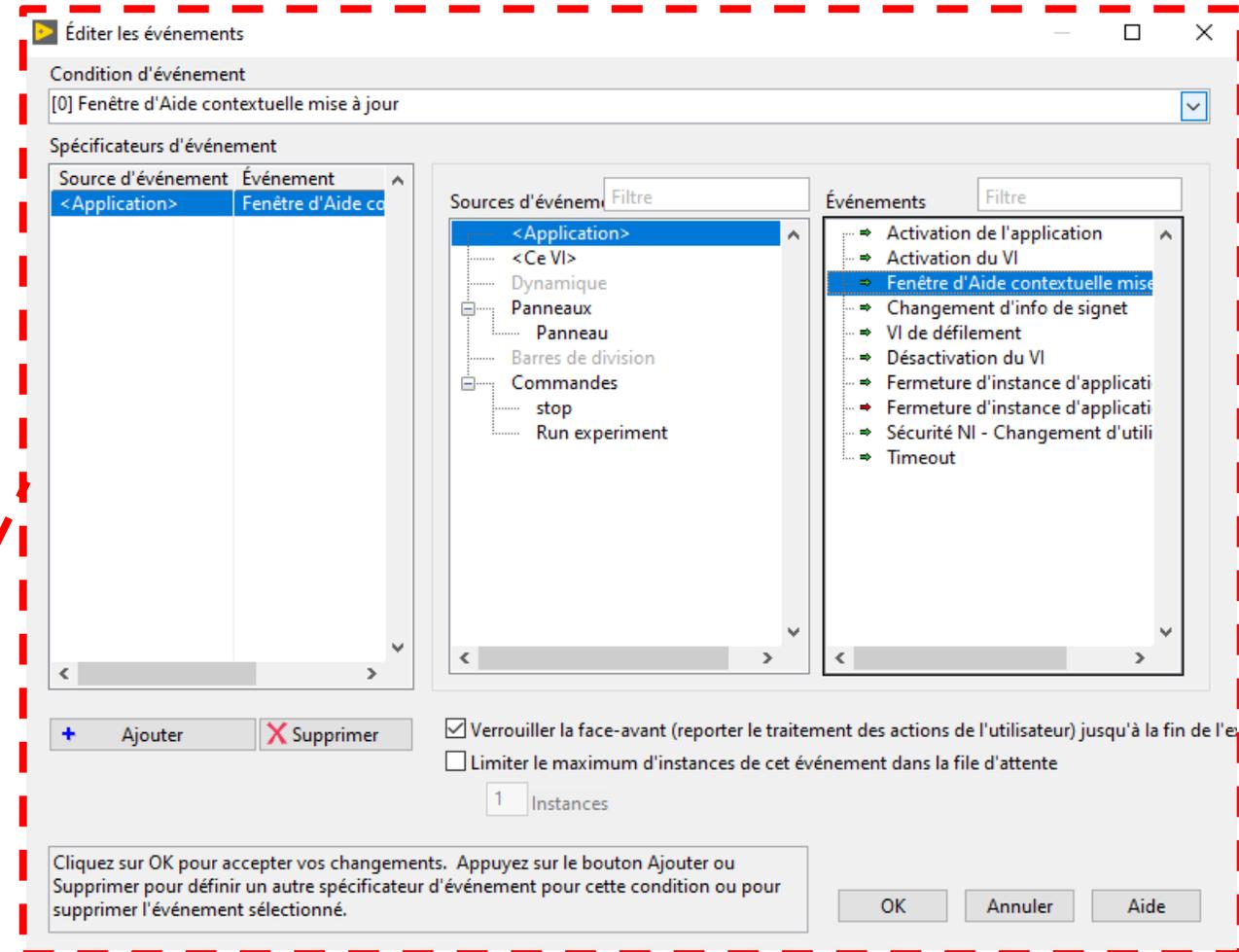
Abonnement statique à des événements

Extrait de l'aide de Labview 2018 sur les **abonnement statique** :

« Vous pouvez configurer une structure Événement afin de spécifier quels événements d'interface utilisateur sur la face-avant d'un VI vous souhaitez gérer dans chaque condition de la structure Événement.

Utilisez la boîte de dialogue Éditer les événements pour configurer une structure Événement afin qu'elle gère un événement auquel vous vous êtes abonné de manière statique.

Remarque Vous ne pouvez pas configurer de manière statique une structure Événement pour qu'elle traite les événements de la face-avant d'un autre VI. »



I. Introduction à la structure événement

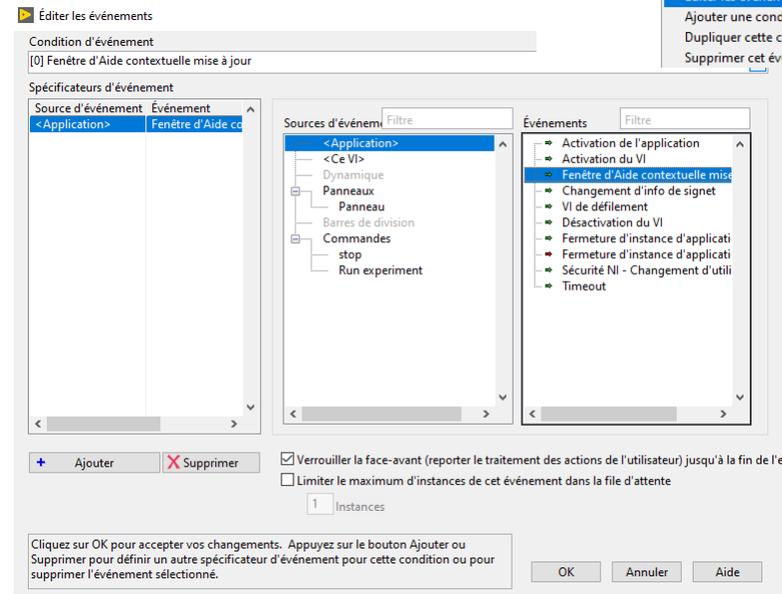
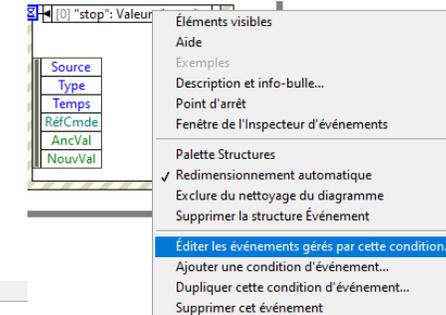
Abonnement statique à des événements

Extrait de l'aide de Labview 2018 sur les **abonnement statique** :

« Vous pouvez configurer une structure Événement afin de spécifier quels événements d'interface utilisateur sur la face-avant d'un VI vous souhaitez gérer dans chaque condition de la structure Événement.

Utilisez la boîte de dialogue Éditer les événements pour configurer une structure Événement afin qu'elle gère un événement auquel vous vous êtes abonné de manière statique.

Remarque Vous ne pouvez pas configurer de manière statique une structure Événement pour qu'elle traite les événements de la face-avant d'un autre VI. »



On ne peut donc pas utiliser les abonnements statiques dans les sous VI!!

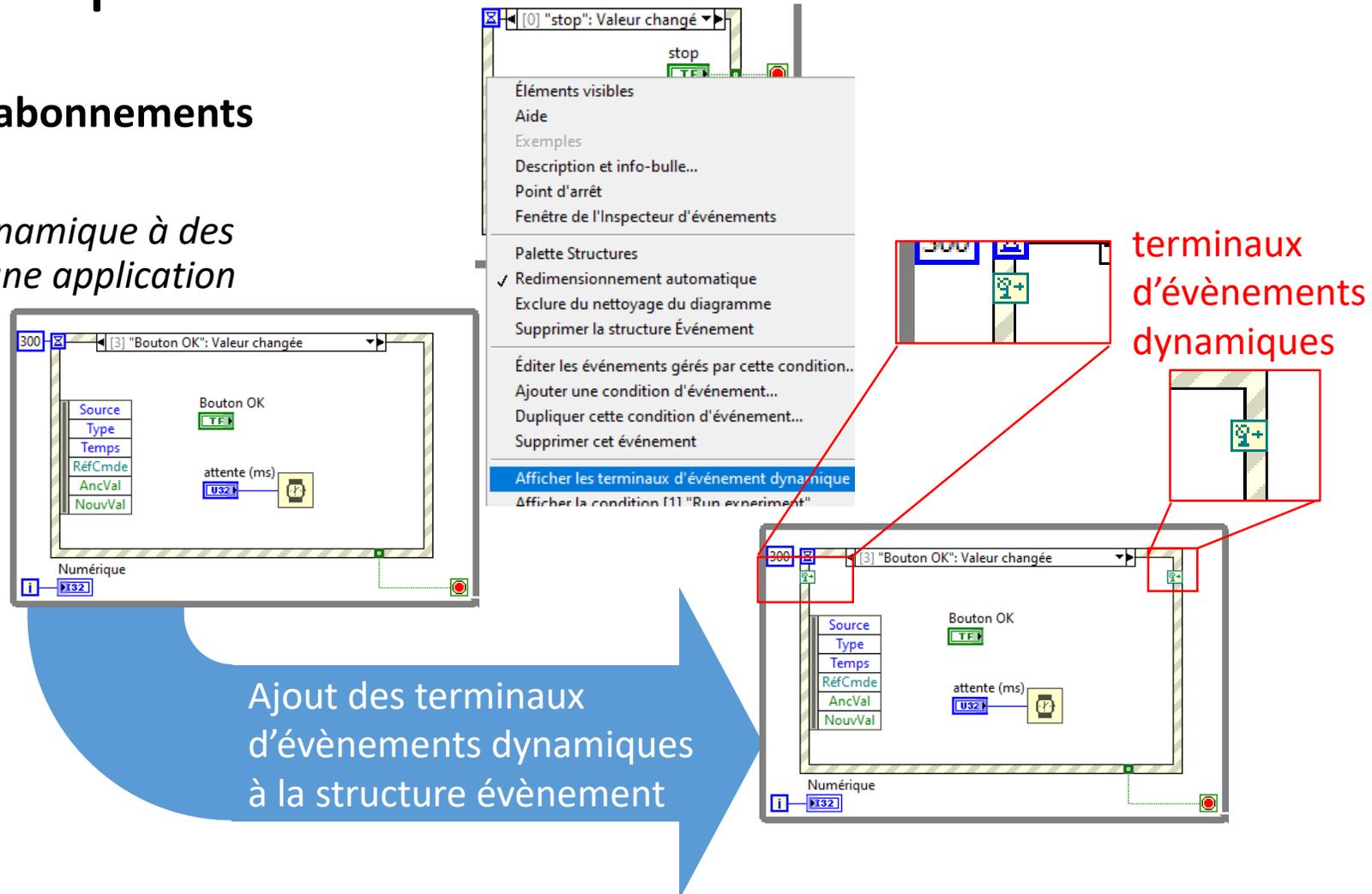
I. Introduction à la structure événement

Abonnement dynamique à des événements

Extraits de l'aide LabView sur les **abonnements dynamiques** :

« Grâce à l'abonnement dynamique à des événements, vous pouvez concevoir une application pour effectuer des changements par programmation. »

=> Il est possible d'utiliser les abonnements dynamiques dans les sous VI!!



Ajout des terminaux d'évènements dynamiques à la structure événement

I. Introduction à la structure événement

Abonnement dynamique à des événements

Exemple type LabView sur les abonnements dynamiques :

« *Dynamic Event Generation.vi* » à retrouver ici :

C:\Program Files (x86)\National Instruments\LabVIEW 2018\examples\Dialog and User Interface\Events

Présentation : Montre comment utiliser la fonction **S'abonner à des événements** pour s'abonner et se désabonner à des événements de manière dynamique lors de l'exécution d'un VI.

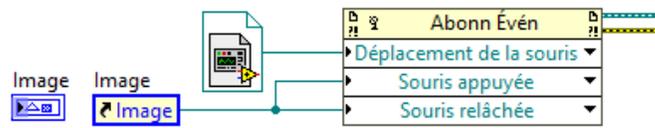
Instructions :

1. Exécutez le VI.
2. Cliquez et faites glisser l'**image**. Notez que l'image se déplace avec la souris et que la valeur **Nombre d'itérations** augmente, ce qui montre que l'événement **Déplacement de la souris** est déclenché.
3. Faites glisser l'image vers le côté inférieur de la fenêtre et arrêtez de la faire glisser. Notez que la valeur Nombre d'itérations ne change plus, étant donné que vous n'êtes plus abonné à l'événement Déplacement de la souris depuis que vous avez arrêté de faire glisser l'image.
4. Cliquez sur **Arrêter** pour arrêter le VI.

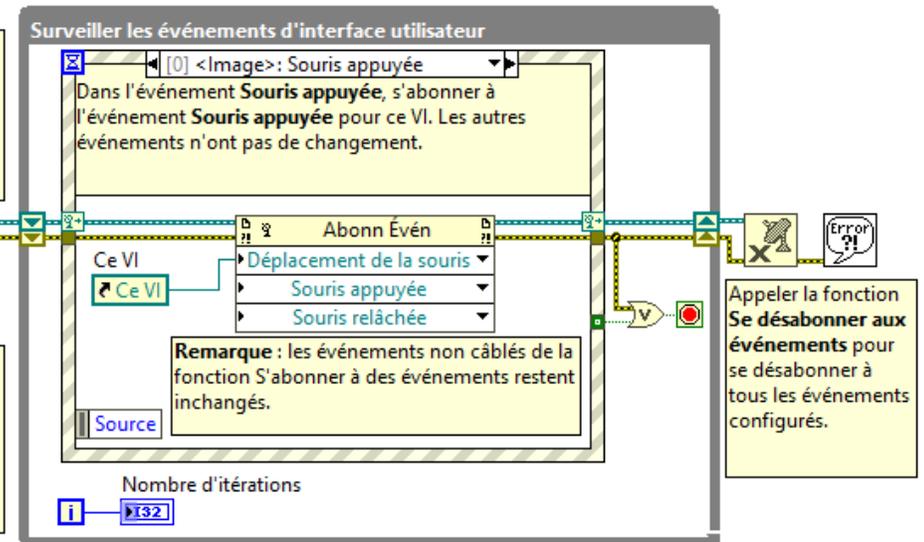


Cet exemple illustre comment s'abonner à des événements de manière dynamique lors de l'initialisation et changer ces événements abonnés durant l'exécution de la structure Événement.

L'événement **Déplacement de la souris** est d'abord abonné à une référence de VI Pas un refnum. Ceci réserve une entrée d'enregistrement mais rien n'est encore enregistré. L'abonnement à Déplacement de la souris sera changé au sein de la structure Événement lors de l'exécution.



Les événements **Souris appuyée** et **Souris relâchée** sont abonnés à l'indicateur Image. Il est aussi possible de câbler des tableaux de références de commande aux événements d'une fonction **S'abonner à des événements** pour permettre à une étape unique d'une structure Événement de s'exécuter quand l'événement spécifié se produit sur n'importe quelle commande spécifiée.



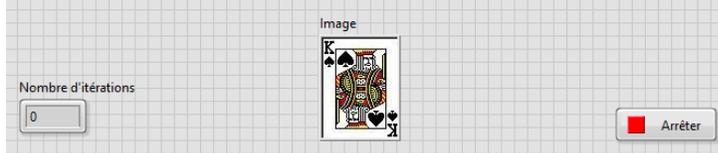
1. Introduction à la structure événement

Abonnement dynamique à des événements

Présentation : Montre comment utiliser la fonction S'abonner à des événements pour s'abonner et se désabonner à des événements de manière dynamique lors de l'exécution d'un VI.

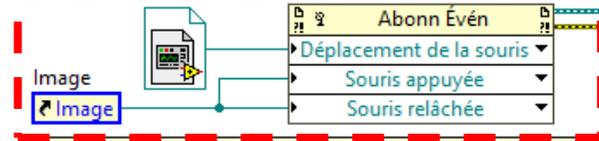
Instructions :

1. Exécutez le VI.
2. Cliquez et faites glisser l'image. Notez que l'image se déplace avec la souris et que la valeur Nombre d'itérations augmente, ce qui montre que l'événement Déplacement de la souris est déclenché.
3. Faites glisser l'image vers le côté inférieur de la fenêtre et arrêtez de la faire glisser. Notez que la valeur Nombre d'itérations ne change plus, étant donné que vous n'êtes plus abonné à l'événement Déplacement de la souris depuis que vous avez arrêté de faire glisser l'image.
4. Cliquez sur Arrêter pour arrêter le VI.

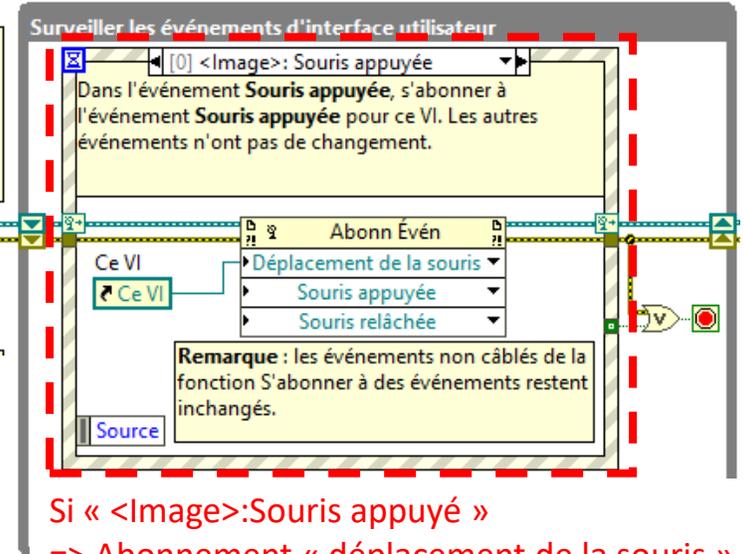


Dans ce VI on peut déplacer la carte pendant l'exécution du vi grâce à une gestion des événements par abonnement dynamique

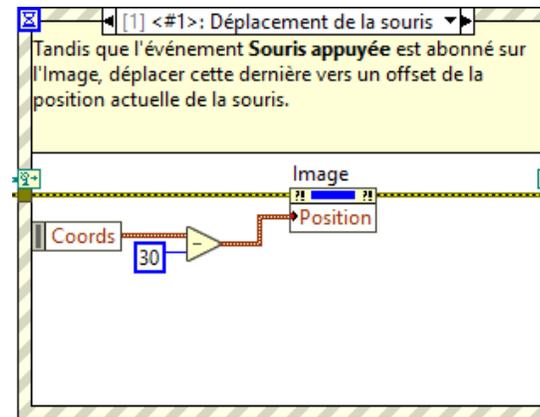
Le VI Pas un refnum. Ceci réserve une entrée de gestion mais rien n'est encore enregistré. L'abonnement de la souris sera changé au sein de la structure événement lors de l'exécution.



3 types abonnements dynamiques créés et « déplacement de la souris » désactivé



Si « <Image>:Souris appuyé » => Abonnement « déplacement de la souris » activé



Puis lors du « Déplacement de la souris », on modifie la position de « Image », la carte du roi de pique

III. De l' v nement statique   dynamique

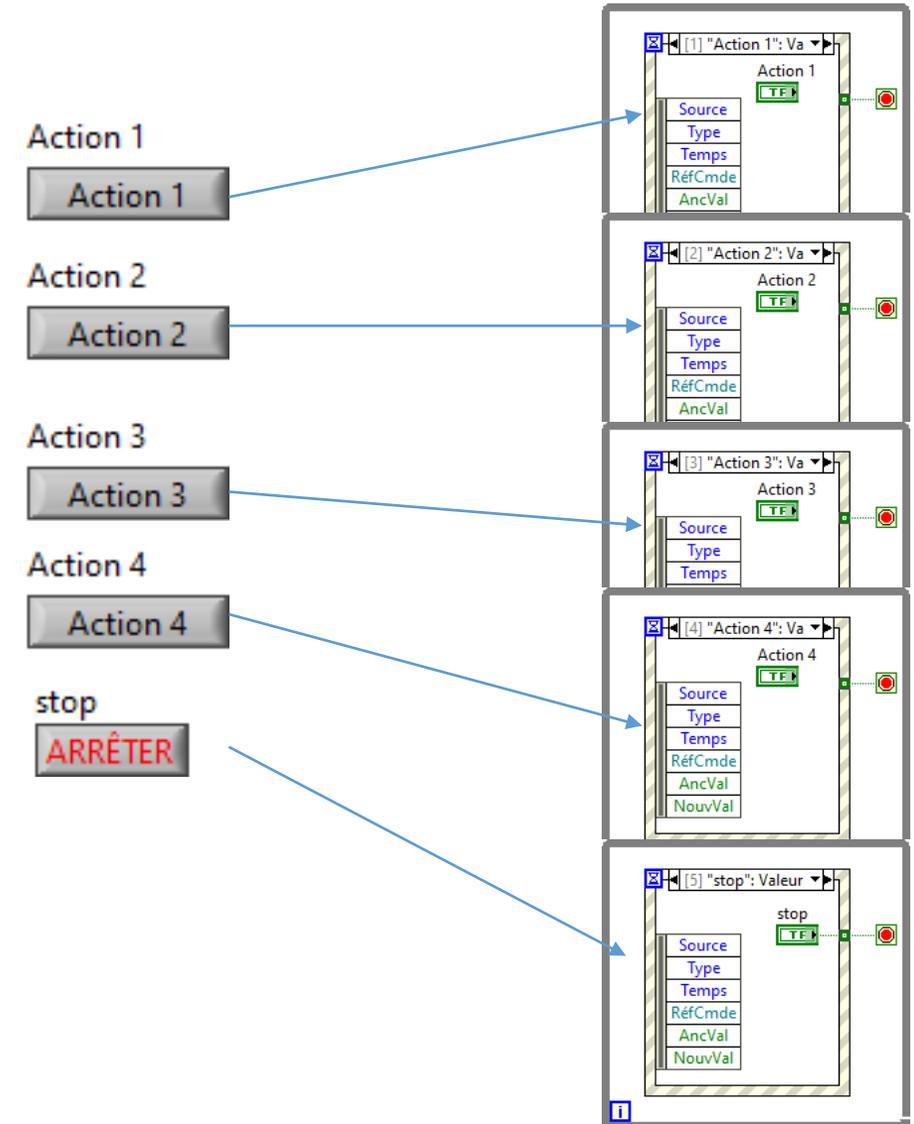
1. Marre de r crire le m me code!
2. Comment j'ai utilis  l'abonnement dynamique
3. Du VI au sous vi

III. De l' v nement statique   dynamique

Marre de r ecrire le m me code!

- Quand on a 6 actions   g rer quasi identiques, ici :
 « Action n » : valeur chang e cela prend du temps inutile via abonnement statique

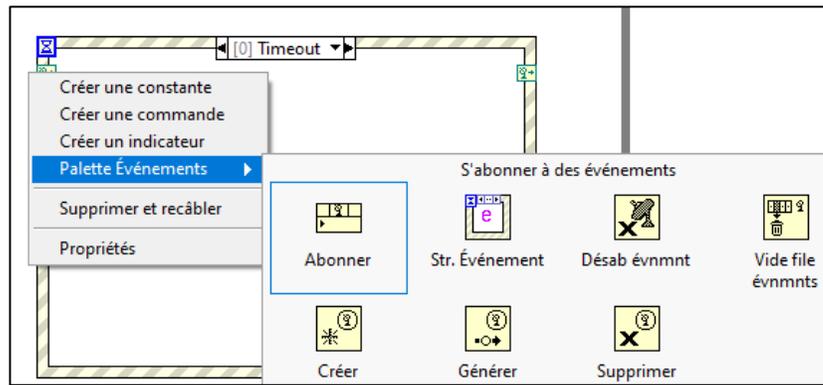
=> n cessit  de cr er un abonnement dynamique



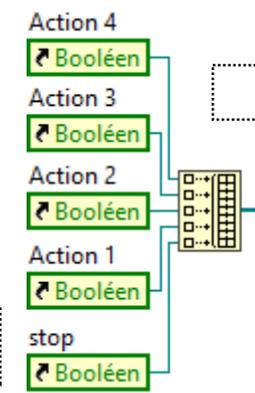
III. De l'évènement statique à dynamique

Marre de réécrire le même code! : Simplification

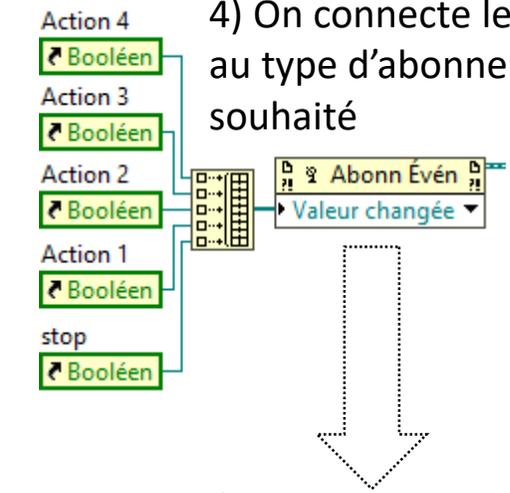
1) On ajoute la fonction s'abonner à des évènements



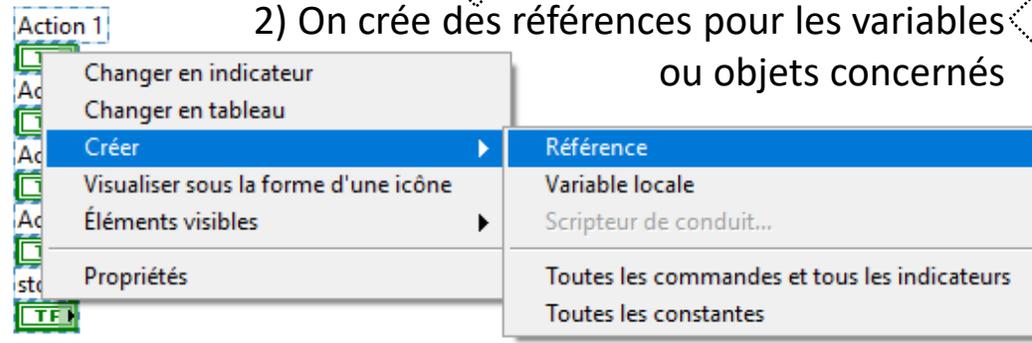
3) On crée un tableau avec ces références



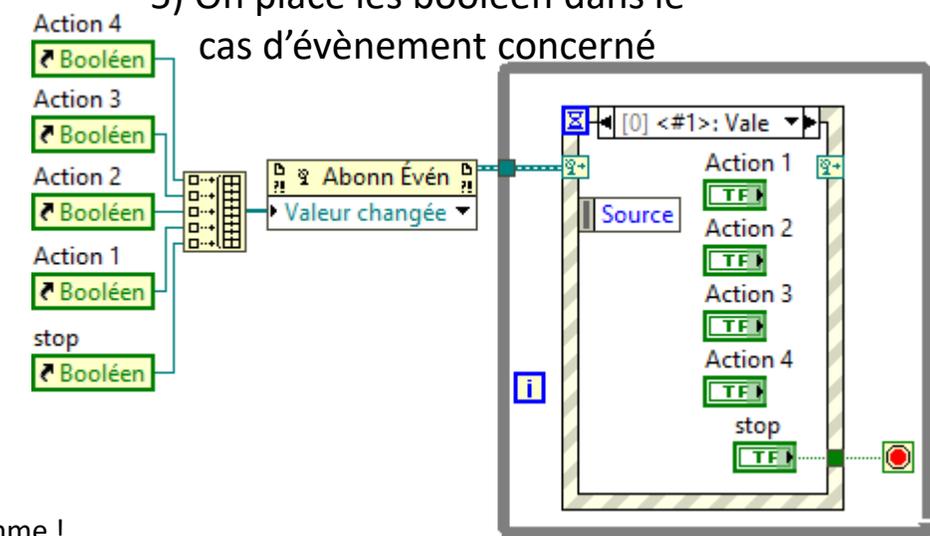
4) On connecte le tableau au type d'abonnement souhaité



2) On crée des références pour les variables ou objets concernés



5) On place les booléen dans le cas d'évènement concerné

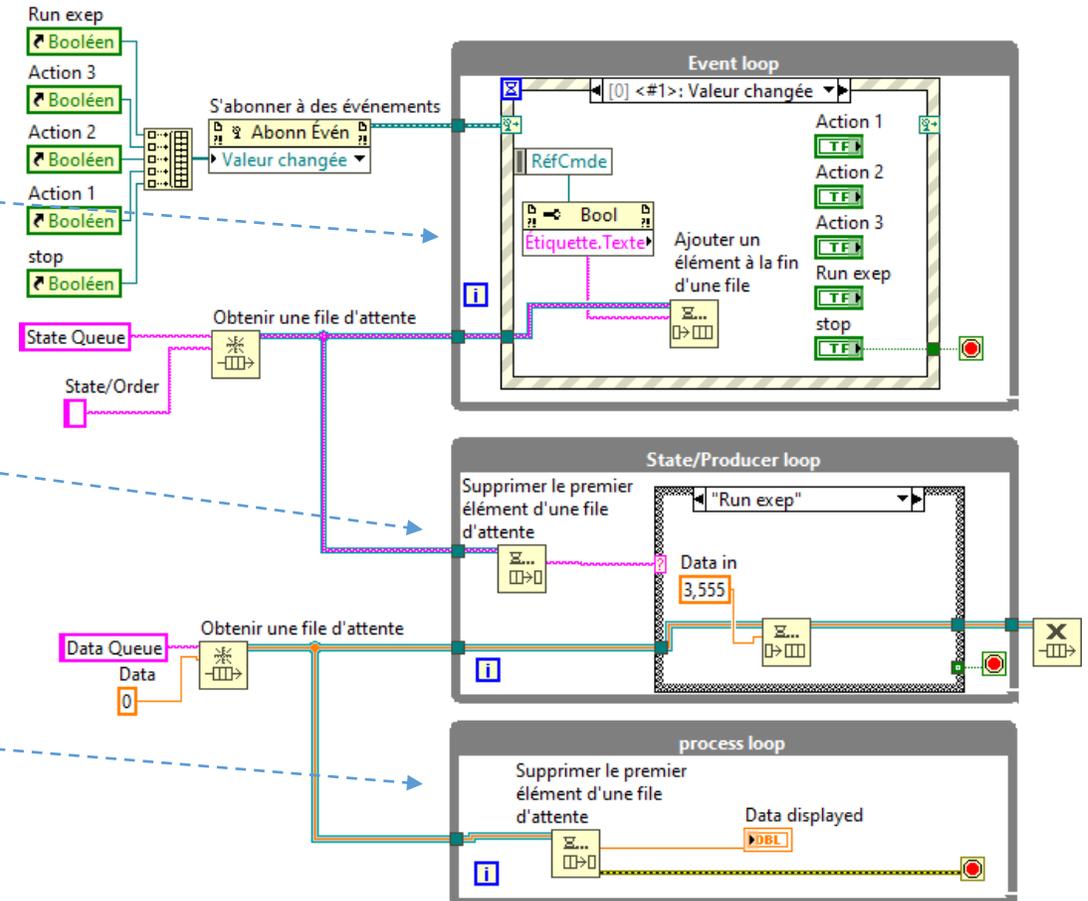


III. De l'événement statique à dynamique

Comment j'ai utilisé l'abonnement dynamique

Au moins 2 voir 3 boucles en parallèles :

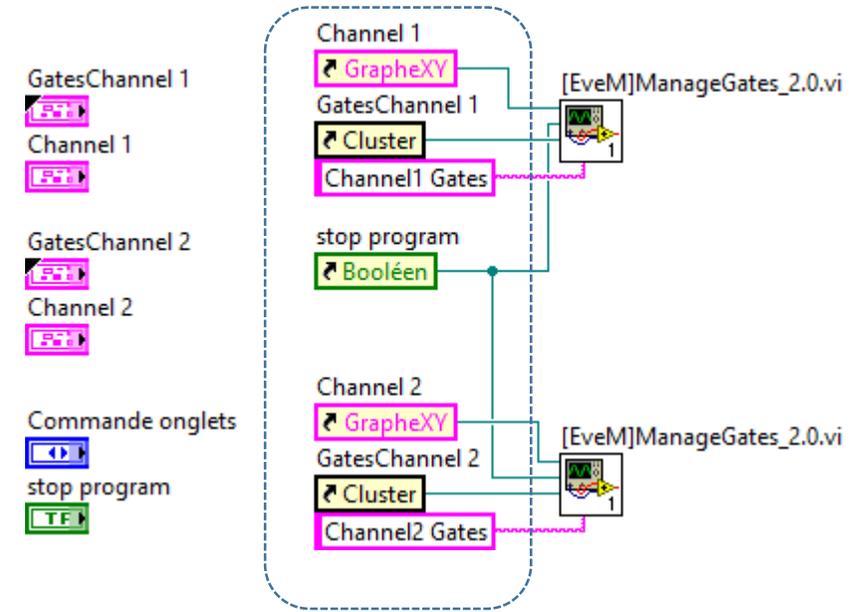
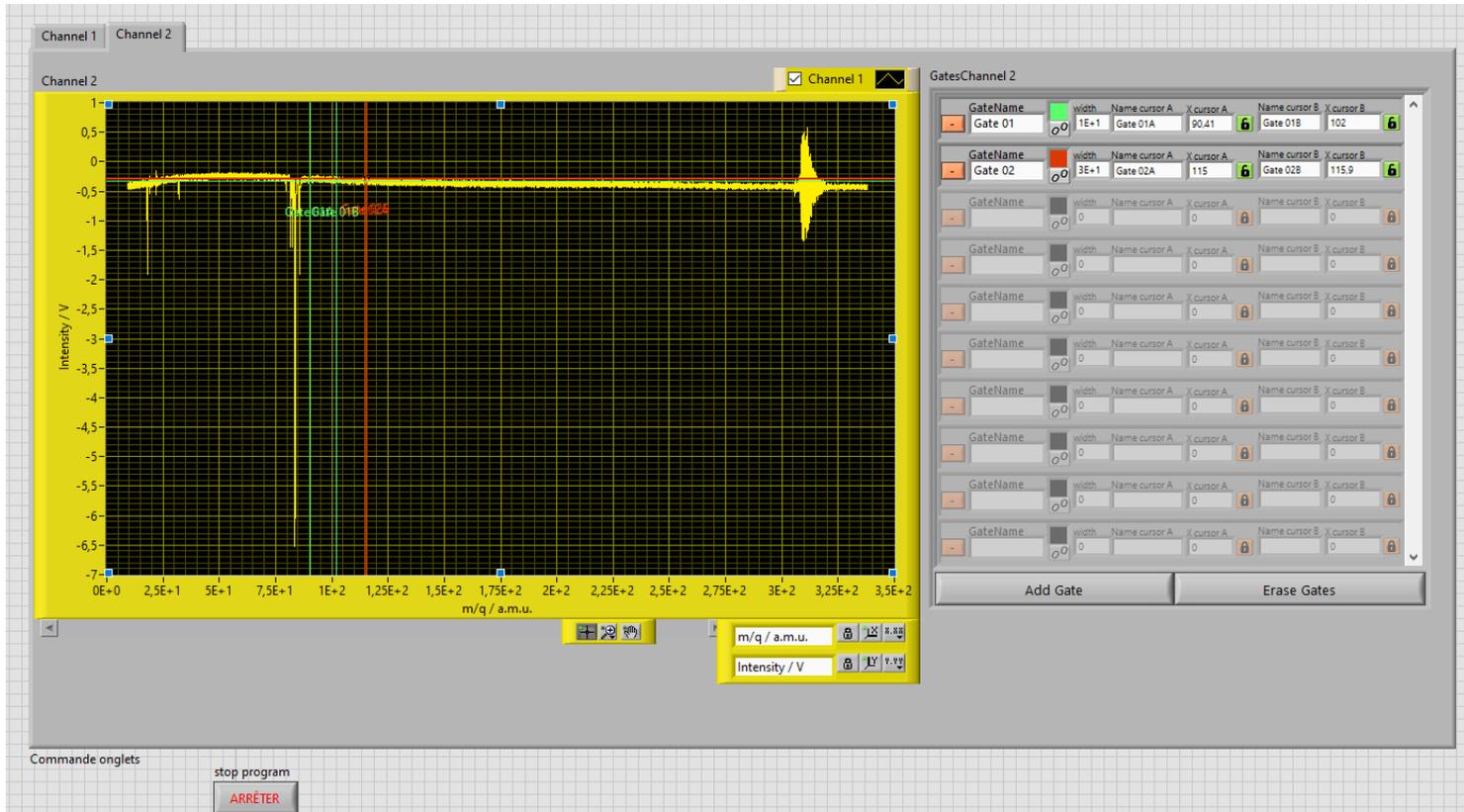
- 1) Boucle évènements : génère des ordres/actions
- 2) Boucle « état/producteur », lit les ordres/actions et change d'état
- 3) Boucle de traitement des données : enregistrement, mise en forme, affichage



III. De l'événement statique à dynamique

Du VI au sous vi : cas pratique VULCAIM

L'application de l'abonnement dynamique au sous-VI peut donner ceci :

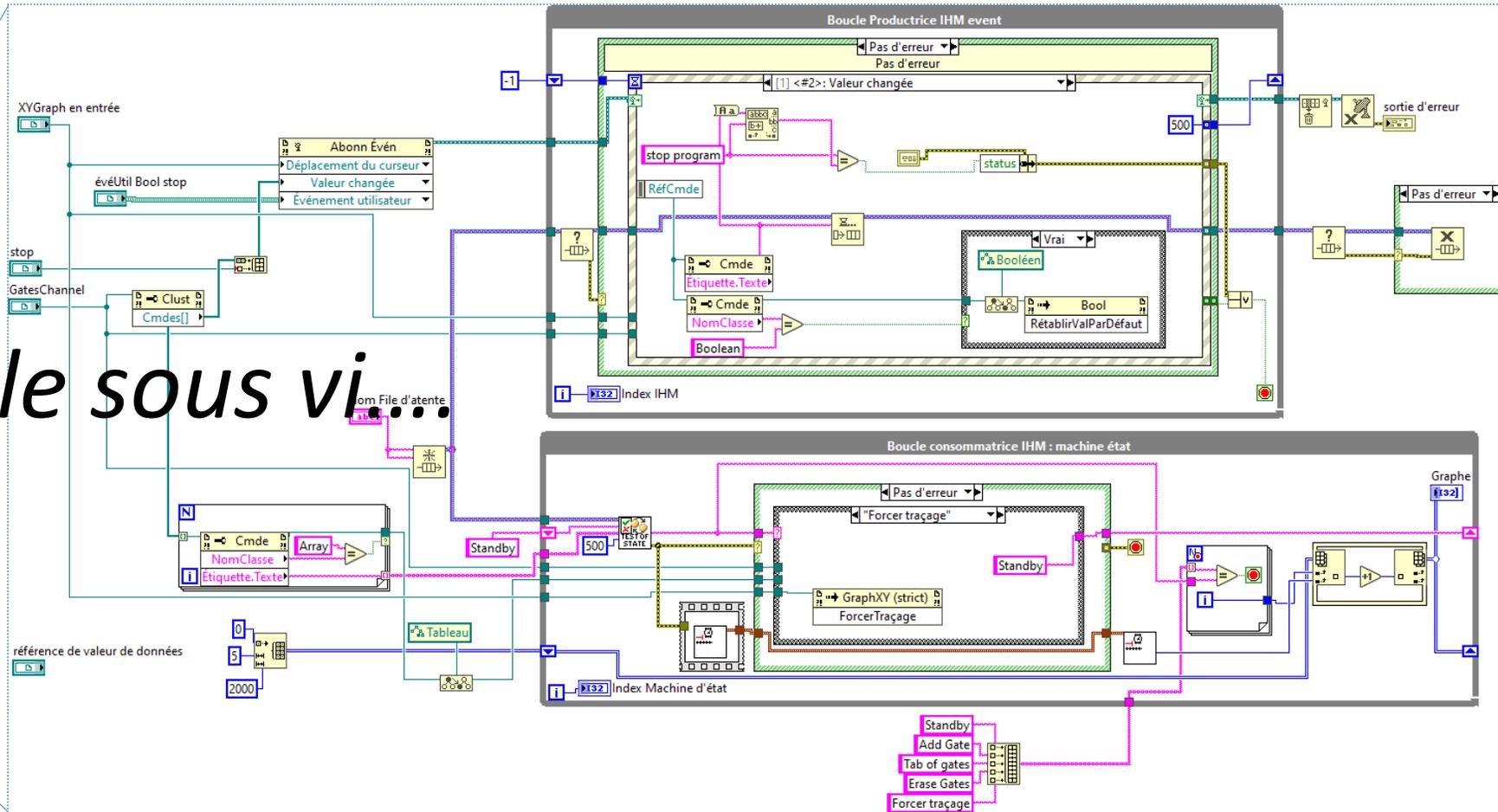
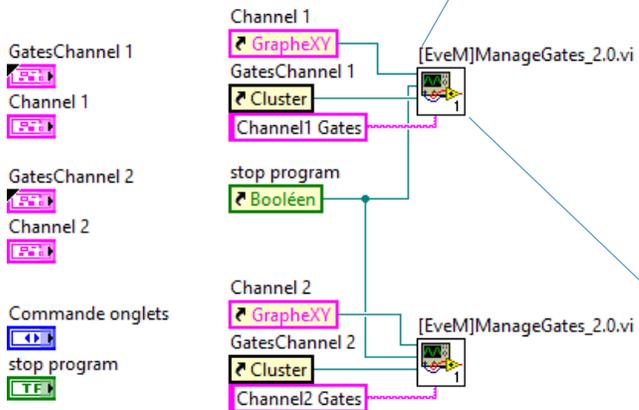


Utilisation des références pour connecter les évènements au sous VI

III. De l'événement statique à dynamique

Du VI au sous vi

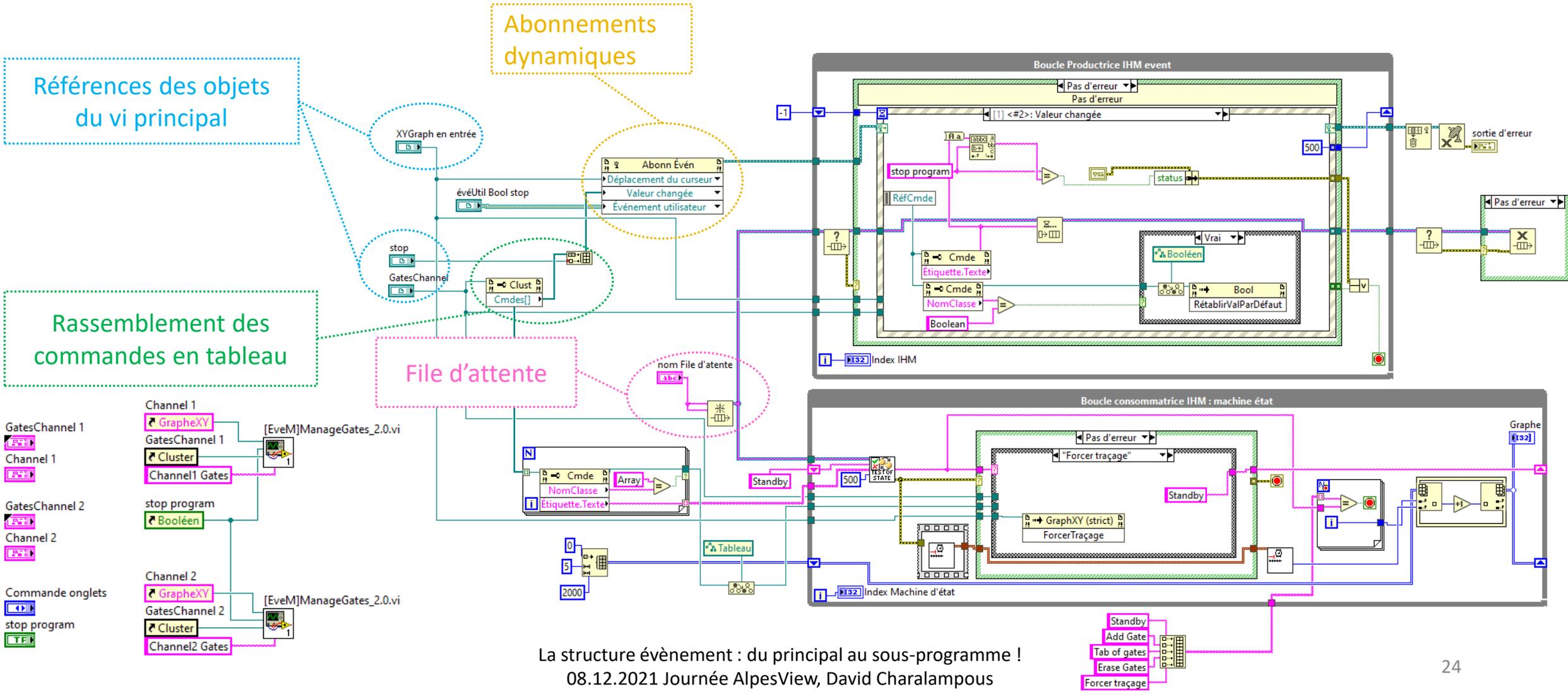
Entrons dans le sous vi...



La structure évènement : du principal au sous-programme !
08.12.2021 Journée AlpesView, David Charalampous

III. De l' v nement statique   dynamique

Du VI au sous vi : initialisation



La structure  v nement : du principal au sous-programme !
 08.12.2021 Journ e AlpesView, David Charalampous

III. De l'évènement statique à dynamique

Du VI au sous vi : initialisation

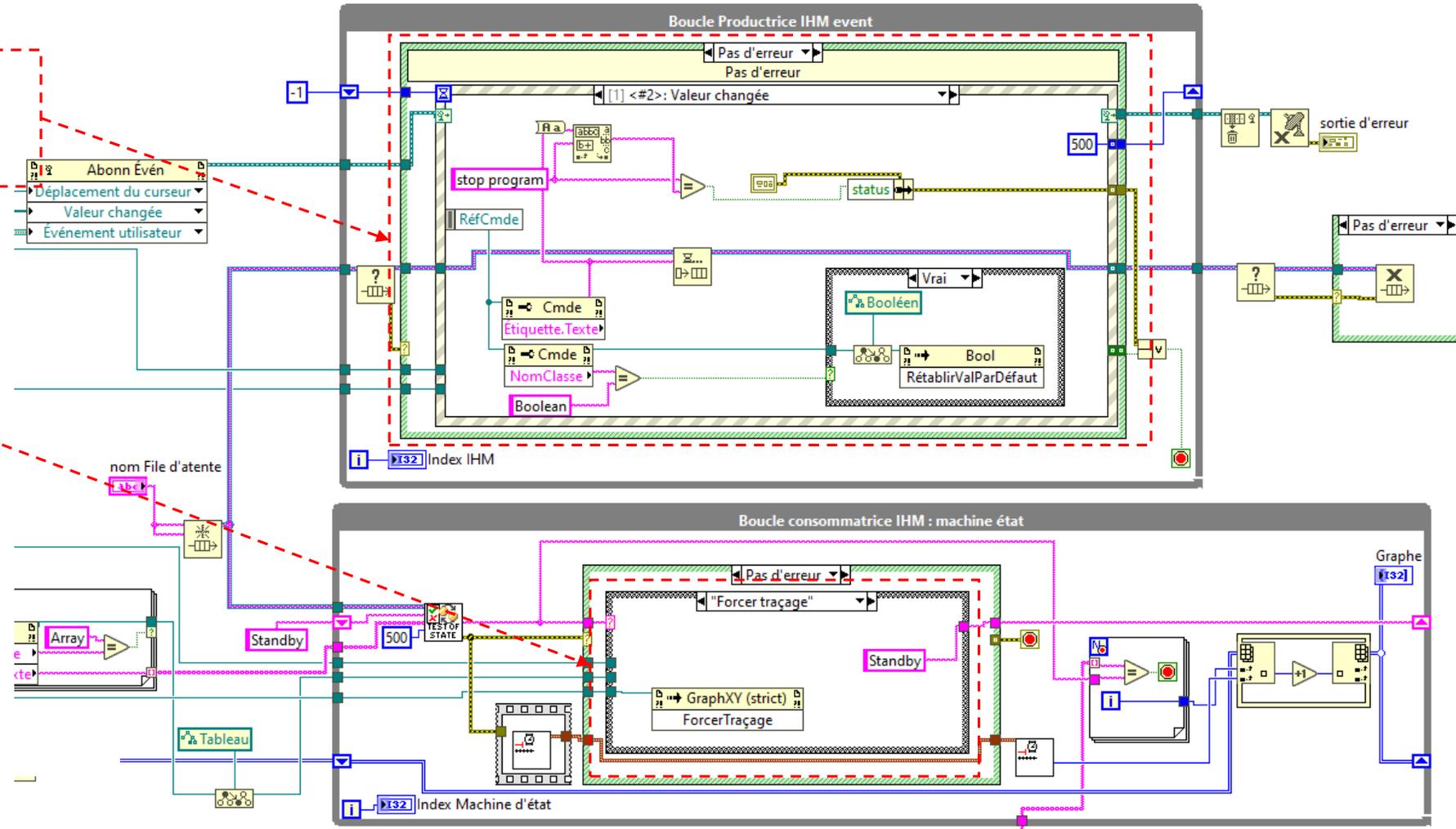
Gestion des évènements :

- Valeur changée
- Déplacement du curseur

Machine d'état :

- Standby
- **Add gate**
- **Erase gate**
- **Tab of gates**
- Forcer traçage

États gérés via file d'attente

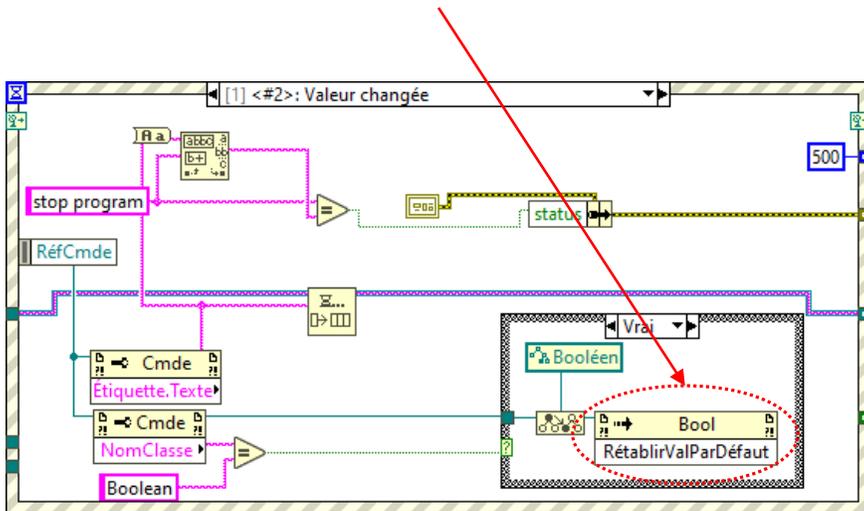


La structure évènement : du principal au sous-programme !
08.12.2021 Journée AlpesView, David Charalampous

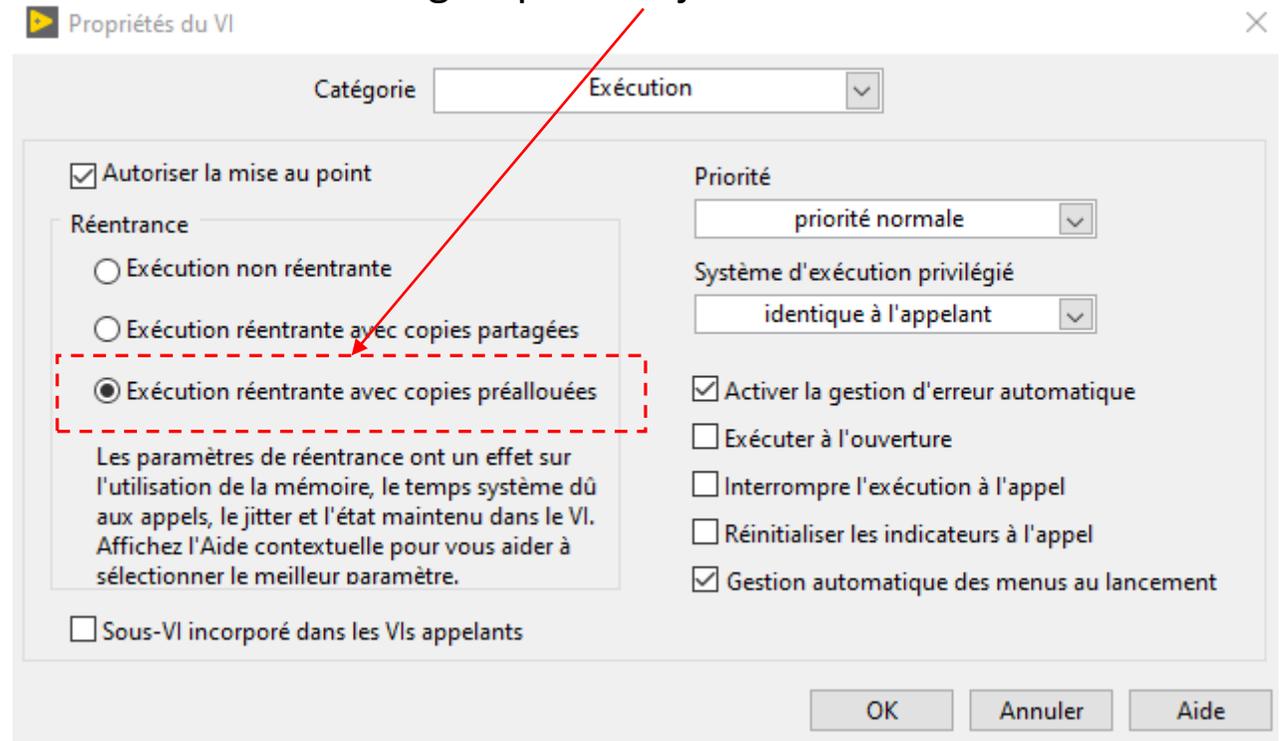
III. De l' v nement statique   dynamique

Du VI au sous vi : 2 astuces n cessaires

N ud de m thode « r tablir par d faut : permet de r armer un bool en   armement



OBLIGATOIRE ICI : l' xecution r entrante avec copie pr allou e
 « clone » chaque sous vi utilis  afin de g rer plusieurs objets ou groupes d'objets distincts

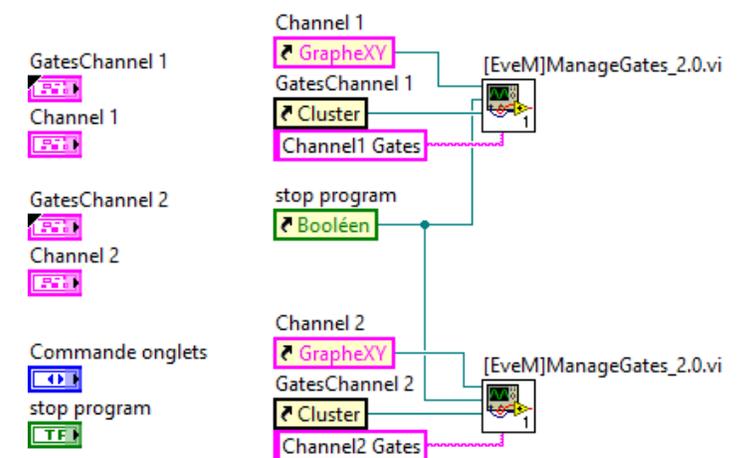


Conclusion

- La structure  v nement apporte plein de fonctionnalit s
- L'abonnement dynamique est indispensable pour une architecture modulaire
- La maitriser l'architecture producteur/consommateur pr f rable
- Demande du temps

- Quelques info compl mentaires en annexe

Avant je n'aurais jamais cru avoir un VI principal si r duit en code :



Remerciements

- D'abord Muriel LAGAUZERE pour l'organisation de cette journ e !!!
- Mon coll gue Chercheur :
B renger GANS
- Mes coll gues du services  lectronique :
Christophe CHARRIERE
Julien VINCENT
- Olivier JOURDAN de Wovalab
- Eric BOBILLIER pour des  changes par mails

R f rences :

- Aide LabVIEW 2018
-  changes avec mes coll gues
- Pr sentation d'Olivier JOURDAN
<https://www.youtube.com/watch?v=i0ajeO2FrEY>
- Echanges mails via liste AlpesView avec Eric BOBILLIER sur l'inspecteur d' v nement

Remerciements

- D'abord Muriel pour l'organisation de cet  v nement !!!
- Mon coll gue Chercheur :
B renger GANS
- Mes coll gues du services  lectronique :
Christophe CHARRIERE
Julien VINCENT
- Olivier JOURDAN de Wovalab

R f rences :

- Aide LabVIEW 2018
-  changes avec mes coll gues
- Pr sentation d'Olivier JOURDAN
<https://www.youtube.com/watch?v=i0ajeO2FrEY>

Merci pour votre attention !!

Annexe

Les 18 commandements de l'évènement

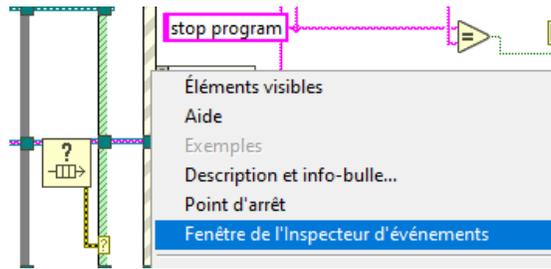
1. Assurez-vous que les [structures Événement sont capables de traiter les événements chaque fois qu'ils se produisent](#).
2. N'oubliez pas de [lire le terminal d'une commande booléenne armée](#) dans sa condition d'évènement Valeur changée.
3. Utilisez une structure Condition pour [gérer les opérations d'annulation pour une commande booléenne armée](#).
4. Soyez également prudent si vous voulez [configurer une condition pour gérer plusieurs événements de type notification](#).
5. [Vous ne pouvez pas configurer une condition pour gérer plusieurs événements de type filtre](#) avec différentes données d'évènement.
6. Si une boucle While qui contient une structure événement se termine sur base de la valeur d'une commande booléenne armée de type bouton stop, n'oubliez pas de [gérer cette commande](#) dans la structure Événement.
7. Considérez [l'utilisation de la fonction Attendre une activité sur la face-avant](#) si vous n'avez pas besoin de surveiller par programmation des objets spécifiques de la face-avant.
8. Les événements interface utilisateur ne s'appliquent qu'à des [interactions utilisateur directes](#).
9. Évitez d'[utiliser des boîtes de dialogue](#) dans une condition Événement comprenant l'évènement de type filtre Souris appuyée ?
10. Évitez de [placer deux structures Événement dans une boucle](#).
11. Évitez d'[utiliser plusieurs structures Événement](#) dans le même VI.
12. Lors de l'utilisation de [l'abonnement dynamique](#), assurez-vous que vous avez une fonction S'abonner à des événements pour chaque structure Événement.
13. Si vous vous abonnez au même événement à la fois sur une classe VI et une classe Commande, LabVIEW génère l'évènement du VI en premier.
14. Lorsque vous [utilisez des commandes de face-avant secondaire](#), le VI de niveau principal qui contient la commande face-avant secondaire se charge de gérer l'évènement.
15. Si vous avez besoin de générer ou de gérer d'autres événements en même temps que l'évènement actuel, envisagez d'utiliser la fonction [Enregistrer un VI Callback d'évènement](#).
16. Faites attention lorsque vous devez choisir entre un [événement de type notification et un événement de type filtre](#). Une condition d'évènement configurée pour traiter un événement de type notification ne peut pas influencer la probabilité ou la manière dont LabVIEW traitera une interaction utilisateur. Si vous voulez modifier la manière dont LabVIEW traite une interaction utilisateur, utilisez l'évènement de type filtre.
17. N'utilisez pas l'évènement notification [Face-avant fermée](#) pour du code d'arrêt de système important, sauf si vous faites le nécessaire pour éviter que le VI ne s'interrompt à la fermeture de la face-avant. Par exemple, assurez-vous que l'application [ouvre une référence](#) au VI avant qu'un utilisateur ne puisse fermer la face-avant. Vous pouvez aussi utiliser l'évènement filtre [Fermeture de face-avant ?](#), qui a lieu avant que la face-avant ne se ferme.
18. Si aucune structure Événement ne s'exécute promptement pour traiter un événement et que le [verrouillage de la face-avant](#) est activé, l'interface utilisateur du VI peut ne pas répondre. Vous devez vous assurer que la structure Événement est dans une boucle qui s'exécute régulièrement pour répondre promptement aux [événements de l'interface utilisateur](#) ou pour désactiver le verrouillage de la face-avant.

Annexe

L'inspecteur d'évènement

Accessible par :

- Le menu en face avant :
Affichage > Fenêtre de l'inspecteur d'évènement
- Clic droit sur structure évènement :

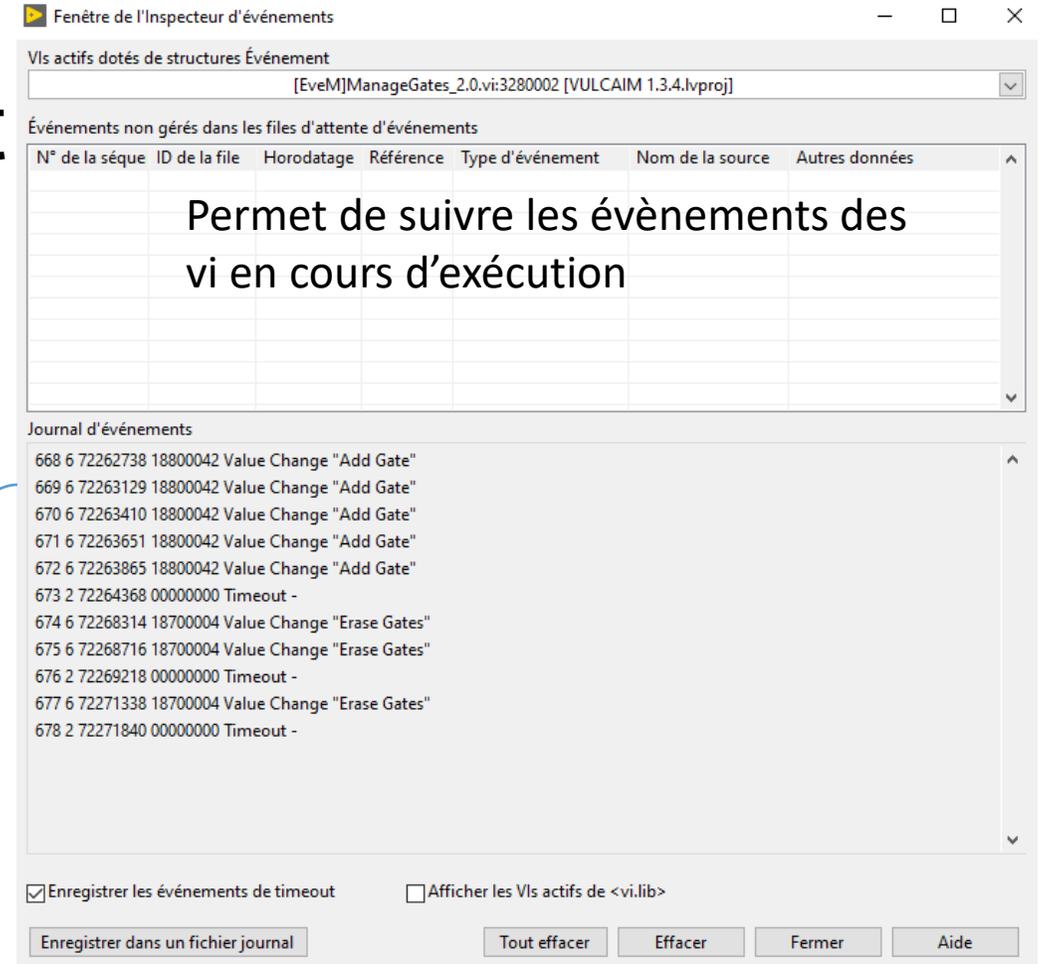


Localisé depuis l'arborescence

..\LabVIEW 2018\resource\Event\EventCB.llb\Event inspector.vi

Ou

..\LabVIEW 2018\resource\dialog\Event Inspector



Permet de suivre les évènements des vi en cours d'exécution

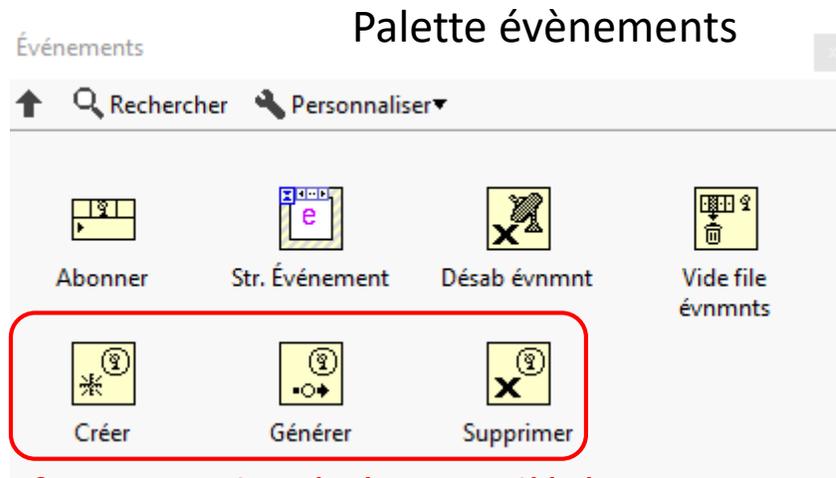
Page Forum NI avec un programme permettant de d'extraire les infos de L'inspecteur d'évènement* :

<https://forums.ni.com/t5/You-May-Not-Know/Programmatically-Inspect-Event-Queues/bc-p/3750124?profile.language=fr>

=> Info donnée par Eric Bobillier sur la mailing liste AlpesVIEW le 30 Août 2021 17:23

Annexe

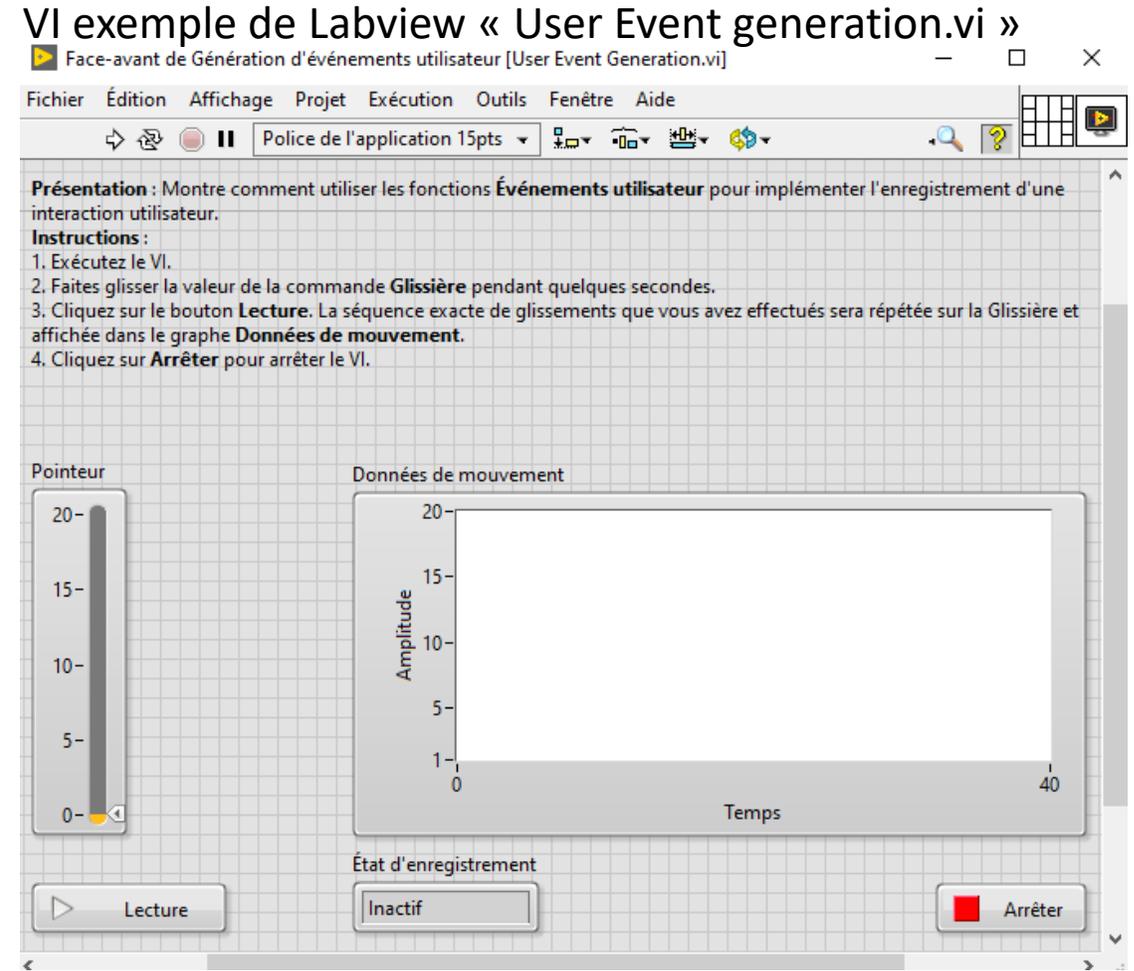
Génération d'événements



fonctions de génération d'évènement :

- Créer : il faut définir un type de variable avec une étiquette
- Générer : on créer l'évènement en envoyant des données
- Supprimer : suppression de la référence évènement

Après création il abonner l'évènement utilisateur à un abonnement dynamique.



Annexe

Génération d'événements



fonctions de génération d'évènement :

- Créer : il faut définir un type de variable avec une étiquette
- Générer : on crée l'évènement en envoyant des données
- Supprimer : suppression de la référence évènement

Après création il s'abonne l'évènement utilisateur à un abonnement dynamique.

